Homework 4: User Interactions

Name: _____

Student ID (usually 112-XXX-XXXX or 113-XXX-XXXX): _____

Advice: When asked to write a code fragment, it is wise to take the code you write by hand and put it in eclipse and experiment with it to make sure it is legal in Java and that it does what is expected.

1.   (10 points; 2 points each) Write a line or two of code (called a code fragment) that uses the variables below, variable(s) that you declare, methods in the Math class, and math operations to perform the given operations. Be sure that you read the method signatures in the Math class carefully (especially the return types) because some are different than what you might expect. If you need to change types, remember that a cast is an option.

```
int frog; // value given elsewhere
int toad; // value given elsewhere
int lizard; // value given elsewhere
double iguana; // value given elsewhere
double skink; // value given elsewhere
```

      a.   Use the ceil operation in the Math class to round the value stored in iguana up to the next larger integer and store the result in frog.

      b.   Use the random method in the Math class to create a random double that is between 0.0 (inclusive) and 50.0 (exclusive) and store the value in skink. This means 0.0 is included but 50.0 is not.

      c.   Use the random method in the Math class to create a random int between 1 (inclusive) and 50 (inclusive). Store this value in toad. Think about what you would need to do to your answer from part b) to make this work.

d.   Find the maximum of the values frog, toad and iguana and store it in skink. You may need to call the max method in the Math class several times.

f.   Use the pow method in the Math class to find the value of 2^{31}. Store this value in lizard.

2. (10 points) Write a code fragment (a few lines of code) that calculates gas mileage from user input data.

The data below is used to keep track of the efficiency of a car. This is usually given in miles per gallon—the number of miles that can be driven on a single gallon of gasoline. The leftmost entry is the date. The next entry on each line is the starting odometer reading (this is the number of miles that the car has been driven). The following entry is the ending odometer reading. The last entry on the line is the number of gallons of gas put into the car. Data in this exact format (but not necessarily the same exact data) is entered into your program using the keyboard.

1/26/23 23489 23794 18.6 Honyota Fantasy Car

If you know there is one line of data in this format but don't know what the data is (do not assume it is the data above—that data is just an example), write some code that will read in the data from the keyboard and print out the miles per gallon for the car, labeled as below. Do not assume that the car name contains exactly three words, there may be more or less.

To calculate the number of miles per gallon you find the number of miles driven (by subtracting the beginning mileage from the ending mileage) and divide by the number of gallons of gas used. Do not worry about the number of decimal points displayed.

1/26/23 Honyota Fantasy Car: 16.397